Noname manuscript No.
(will be inserted by the editor)

Rate-Distortion Analysis of Multiview Coding in a DIBR Framework

Boshra Rajaei · Thomas Maugey · Hamid-Reza Pourreza · Pascal Frossard

the date of receipt and acceptance should be inserted later

Abstract Depth image based rendering techniques for multiview applications have been recently introduced for efficient view generation at arbitrary camera positions. Encoding rate control has thus to consider both texture and depth data. Due to different structures of depth and texture images and their different roles on the rendered views, distributing the available bit budget between them however requires a careful analysis. Information loss due to texture coding affects the value of pixels in synthesized views while errors in depth information lead to shift in objects or unexpected patterns at their boundaries. In this paper, we address the problem of efficient bit allocation between textures and depth data of multiview video sequences. We adopt a rate-distortion framework based on a simplified model of depth and texture images. Our model preserves the main features of depth and texture images. Unlike most recent solutions, our method permits to avoid rendering at encoding time for distortion estimation so that the encoding complexity is not augmented. In addition to this, our model is independent of the underlying inpainting method that is used at decoder. Experiments confirm our theoretical results and the efficiency of our rate allocation strategy.

Keywords depth image based rendering · multiview video coding · rate allocation · rate-distortion analysis

1 Introduction

Three-dimensional video coding is a research field that has witnessed many technological revolutions in the recent years. One of them is the significant improvement in the capabilities of camera sensors. Nowadays, high quality camera sensors that capture color and depth information are easily accessible [1]. Obviously this brings important modifications in the data that the 3D transmission systems have to process. A few years ago, transmission systems used disparity to improve the compression performance [2,3]. Nowadays, 3D systems rather employ depth information to improve the quality experience by, for example, increasing the number of views that could be displayed at the receiver side [4,5]. This is possible because of depth image based rendering (DIBR) techniques [6,7] that project one reference image onto virtual views using depth as geometrical information. Figure 1 shows the overall structure of a DIBR multiview coder that is also considered in this paper. It includes the following steps: first, the captured views in addition to their corresponding depth maps are coded at bit rates assigned by a rate allocation method. Then the coded information are transmitted to the decoder. Finally, at the decoder the reference views are decoded and virtual views are synthesized using the depth information. View synthesis consists of two parts; projection into the virtual view location using closest reference

B. Rajaei
Ferdowsi University of Mashhad, Iran
Sadjad Institute of Higher Education, Mashhad, Iran
Tel.: +98-511-6029000
E-mail: boshra.rajaei@stu-mail.um.ac.ir

T. Maugey
Signal Processing Laboratory (LTS4), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
E-mail: thomas.maugey@epfl.ch

H.-R. Pourreza
Ferdowsi University of Mashhad, Iran
E-mail: hpourreza@um.ac.ir

P. Frossard
Signal Processing Laboratory (LTS4), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
E-mail: pascal.frossard@epfl.ch

views and inpainting for filling the holes [8,9] or pixels that remain undetermined after projection.

DIBR techniques offer new possibilities but also impose new challenges. One of the important questions relies in the effect of depth compression on the view synthesis performance [10]; in particular, for a given bit budget R, what is the best allocation between depth and texture data or in other words, how can we distribute the total bitrate between color and geometrical information in order to maximize the rendering quality? It is important to note that the quality of the rendered view is of interest here, and not the distortion of depth images [10,11]. This renders the problem of rate allocation particularly challenging.

The rate allocation problem has been the topic of many researches in the past few years. Allocating a fixed percentage of total budget to the texture and depth data is probably the simplest allocation policy in the DIBR coding methods [12,13,14]. More efficient methods have however been proposed recently, and we discuss them in more details below.

Starting from the current multiview coding (MVC) profile of H.264/AVC [15,16,3], we should mention that MVC uses the distortion of depth maps to distribute the available bit budget between texture and depth images. A group of papers try to improve MVC by taking into account depth properties. In [17], authors suggest a preprocessing step based on an adaptive local median filter to enhance spatial, temporal and inter-view correlations between depth maps and consequently, improve the performance of MVC. Using the correlation between reference views, the work in [18] skips some depth blocks in the coding and hence, reduces the required bit budgets for coding depth maps. Other methods try to estimate at encoder the distortion of virtual views, which then replaces the depth map distortion in the mode decision step of the MVC method [15]. In [19], the authors provide an upper bound for virtual view distortion that is related to the depth and texture errors and the gradients of the original reference views. Another upper bound for rendered view distortion proposed when encoder has access to the original intermediate views at the encoder [20]. In [21], the algorithm calculates the translation error induced by depth coding and then estimates the rendered view distortion from the texture data. In a similar approach, the work in [22] models the distortion at each pixel of a virtual view, including the pixels in occluded regions. These methods only try to improve the current MVC profile and without modeling the distortion rate behavior, they can not be used as general solutions for the rate allocation problem.

Beside improving the current MVC allocation policy, other papers build a complete rate-distortion model to solve the rate allocation problem of distributing total bit budget between texture and depth data in a DIBR multiview coder [23,24,25,26,27]. For example, assuming independency between depth and texture errors, the work in [23] proposes a DR function to find the optimal allocation in a video system with one reference and one virtual view. A region-based approach for estimating the distortion at virtual views is proposed in [25]. Here, the allocation scheme is an iterative algorithm that needs to render one virtual view at every iteration for parameter initialization. This is very costly in terms of computational complexity. Along the same line of research, we also notice the rate allocation and view selection method proposed in [26]. In this work, the authors first provide a cubic distortion model for synthetic views; they estimate the model coefficients by rendering at least one intermediate view between each reference camera views. Then, using this distortion model, a DR function is formulated and a modified search algorithm is executed to simplify rate allocation. Finally, a DR function is provided for a layer-based depth coder in [27]. The main drawbacks in the above allocation schemes reside in the rendering of at least one virtual view at encoding time and in the construction of DR functions that are view dependent. Rendering at encoder side dramatically increases the computational complexity of the coder and is therefore not acceptable for realtime applications. In addition for view rendering at arbitrary camera positions, multiview systems require rate allocation strategies that work independently of reference and virtual view numbers and exact positions.

In this paper, we propose a novel DR model to solve the rate allocation problem in DIBR coding with arbitrary number of reference and virtual views and without rendering at the encoder side. Inspired by [28,29,30], we first simplify different aspects of a multiview coder and keeping only the main features. In particular, we make simple models for depth and texture coders, camera setup and scene under observation. Then, using a rate-distortion framework, a DR function is calculated and eventually is used for optimizing the allocation problem in multiview coding. An important property of our allocation method is that, we do not consider the inpainting step for virtual view synthesis at the decoder. There are two reasons for this choice: first, we want to design an allocation strategy that is independent of the actual inpainting method; second, we focus on the effect of view projections, which is mostly related to the geometry of the scene. Experimental results show that our model-based rate allocation method is efficient for different system configurations. The approach proposed in this paper has low complexity but provides a distor-

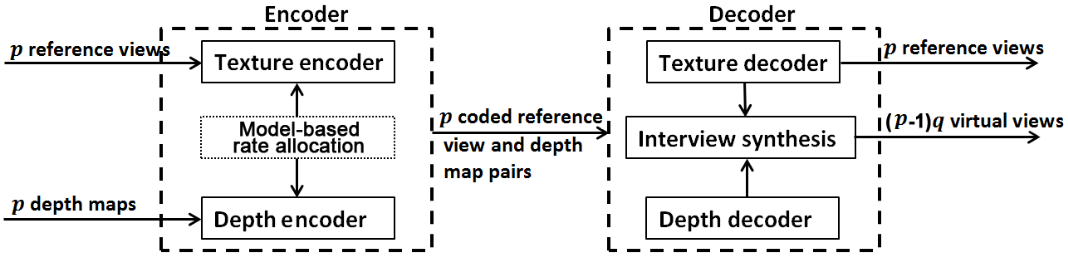

Fig. 1 A DIBR multiview coder structure with p reference cameras and q equally spaced virtual views between each two reference views.

tion that is not far from optimum, and in particular it outperforms a priori rate allocation strategies that are commonly used in practice.

The organization of this paper is as follows. Next section clarifies some notations, camera and scene model and rate-distortion framework as it is used in Section 3 for calculation of our allocation model. Section 4 addresses a few optimization issues. Finally, Section 5 includes the details of our experimental results, parameter values and comparison to other allocation strategies.

2 Framework and model

In this section we define a few preliminary concepts that are used in our rate distortion study. Our main focus is the problem of distributing the available bit rate between several reference views and depth maps in a DIBR multiview video application, such that the distortion over all reference and rendered views at the decoder is minimized. In particular we are interested in constructing a rate-distortion for rate allocation without explicit view synthesis at the encoder.

We first construct a rate-distortion model for a typical wavelet-based texture coder and a simple quantized-based depth map coder, along with a simple model of scene. We present below some general notations and the wavelet framework. Then we describe our rate-distortion analysis framework, our model of the scene and of the camera.

2.1 Notation

Let $\mathbb{R}$ be the set of real numbers. The L_2-norm of a function $f : \mathbb{R}^2 \to \mathbb{R}$ is defined as $\|f\|_2 = \left(\iint f^2(t_1,t_2)dt_1dt_2\right)^{\frac{1}{2}}$. Then, $L_2(\mathbb{R}^2)$ is the set of all functions $f : \mathbb{R}^2 \to \mathbb{R}$ with a finite L_2-norm. The angle bracket represents the inner product of two functions in this space, i.e., for $f, g \in L_2(\mathbb{R}^2)$ we have

$$\langle f, g\rangle = \iint f(t_1,t_2)g(t_1,t_2)dt_1dt_2.$$

Then, let $\phi : \mathbb{R} \to \mathbb{R}$ and $\psi : \mathbb{R} \to \mathbb{R}$ be the univariate scaling and wavelet functions of an orthonormal wavelet transform, respectively [31]. The shifted and scaled forms of these functions are denoted by $\psi_{s,n}(t) = 2^{s/2}\psi(2^s t - n)$ and $\phi_{s,n}(t) = 2^{s/2}\phi(2^s t - n)$, where $s, n \in \mathbb{Z}$ are respectively the scaling and shifting parameters and $\mathbb{Z}$ is the set of integer numbers. The most standard construction of two-dimensional wavelets relies on a separable design that uses $\Psi^1_{s,n_1,n_2}(t_1,t_2) = \phi_{s,n_1}(t_1)\psi_{s,n_2}(t_2)$, $\Psi^2_{s,n_1,n_2}(t_1,t_2) = \psi_{s,n_1}(t_1)\phi_{s,n_2}(t_2)$, and $\Psi^3_{s,n_1,n_2}(t_1,t_2) = \psi_{s,n_1}(t_1)\psi_{s,n_2}(t_2)$ as the bases. It is proved in [31] that separable wavelets provide an orthonormal basis for $L_2(\mathbb{R}^2)$. Therefore, any function $f \in L_2(\mathbb{R}^2)$ can be written as

$$f(t_1,t_2) = \sum_{s,n_1,n_2}\sum_{i=1}^{3} C^i_{s,n_1,n_2}\Psi^i_{s,n_1,n_2}(t_1,t_2),$$

where, for every $s, n_1, n_2 \in \mathbb{Z}$,

$$C^i_{s,n_1,n_2} = \langle f, \Psi^i_{s,n_1,n_2}\rangle,\ i = 1,2,3.$$

Practically, the wavelet transform defines a scale s_0 as the largest scale value. If we call C^i_{s,n_1,n_2} high frequency bands, at s_0 we thus have only one low frequency band $\langle f, \Phi_{s_0,n_1,n_2}\rangle$, where $\Phi_{s_0,n_1,n_2}(t_1,t_2) = \phi_{s_0,n_1}(t_1)\phi_{s_0,n_2}(t_2)$.

2.2 Scene and camera configuration model

We use a very simple model of the scene in our analysis we consider foreground objects with arbitrary shapes and flat surfaces on a flat background[1]. Additionally, even though a real scene is 3D, our model is a collection of 2D images as we consider projections of the 3D scene into cameras 2D coordinates.

Let $\mathcal{H}^Q(\mathbf{\Omega})$ be the space of 2D functions, $f : \mathbb{R}^2 \to \mathbb{R}$, on the interval $[0,1]^2 \subset \mathbb{R}^2$, where Q is the number of foreground objects and $\mathbf{\Omega} = \{\Omega_i, i = 0,\ldots,Q-1\}$

[1] The extension of our analysis to the scenes with C^α regular surfaces is straightforward.

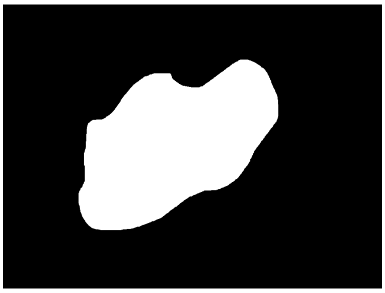

Fig. 2 A sample function in $\mathcal{H}^1(\boldsymbol{\Omega})$.

denotes the foreground objects. We define $f \in \mathcal{H}^Q(\boldsymbol{\Omega})$ as

$$f(t_1,t_2) = \begin{cases} 1, & \text{if } \exists i : (t_1,t_2) \in \Omega_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Our RD analysis is performed on $\mathcal{H}^1(\boldsymbol{\Omega})$ where $\boldsymbol{\Omega} = \{\Omega_0\}$. The extension to multiple foreground objects follows naturally. For the sake of clarity, we skip superscript notation and represent this class by $\mathcal{H}(\Omega)$. Figure 2 shows a sample function from $\mathcal{H}(\Omega)$. This figure shows one arbitrary shape foreground object on a flat background as it is projected into a 2D camera plane.

In addition to our simple scene model, we describe now our camera configuration model. Let us denote as $\mathcal{B}_q^p(\mathcal{P})$ a configuration with p reference cameras and q equally spaced intermediate views between each two reference views. Then, $\mathcal{P}$ is the set of intrinsic and extrinsic parameters for reference and virtual cameras. It is defined as $\mathcal{P} = \{(A_i, R_i, T_i) : i = 0, \dots, p-1\} \cup \{(A'_j, R'_j, T'_j) : j = 0, \dots, (p-1)q\}$, where A_i and R_i are respectively the intrinsic and rotation matrices of ith reference camera and T_i is its corresponding translation vector. The same parameters for virtual cameras are given by A'_j, R'_j and T'_j. Figure 1 shows a multiview coder that corresponds to a $\mathcal{B}_q^p(\mathcal{P})$ configuration. In this paper, we consider that a texture image and a depth map are coded and are sent to the decoder for each reference view. In our camera configuration $\mathcal{B}_q^p(\mathcal{P})$, we have p pairs of texture images and depth maps to be coded at each time slot. The number of coded views is given by system design criteria or rate-distortion constraints [26].

2.3 Rate-distortion framework

Let us define three classes of signals $\mathcal{T} \subset L_2(\mathbb{R}^2)$, $\mathcal{V} \subset L_2(\mathbb{R}^2)$ and $\mathcal{D} \subset L_2(\mathbb{R}^2)$ as reference images, virtual views and depth maps, respectively. Then, define $\mathcal{F}$ as the class of all $f = \{(t_i, d_i) : t_i \in \mathcal{T}, d_i \in \mathcal{D}, i = 0, \dots, p-1\}$ and similarly, $\mathcal{G}$ as the class of all $g = \{(t_i, v_j) : t_i \in \mathcal{T}, v_j \in \mathcal{V}, i = 0, \dots, p-1, j = 0, \dots, q-1\}$. Here, $\mathcal{F}$ represents all the coded data and $\mathcal{G}$ indicates the set of all reference and virtual views that are reconstructed at the deocer.

A typical multiview video coding strategy consists of at least three building blocks namely, encoder, decoder and rendering algorithm. Consider a texture encoding scheme $\mathcal{E}_{\mathcal{T}} : \mathcal{T} \to \{1, 2, \dots, 2^{R_{\mathcal{T}}}\}$ and similarly a depth encoding scheme $\mathcal{E}_{\mathcal{D}} : \mathcal{D} \to \{1, 2, \dots, 2^{R_{\mathcal{D}}}\}$, where $R_{\mathcal{T}} = \sum_{i=0}^{p-1} R_{t_i}$ and $R_{\mathcal{D}} = \sum_{i=0}^{p-1} R_{d_i}$ are the total number of allocated bits to texture and depth information, respectively. This represents a total rate $R = R_{\mathcal{T}} + R_{\mathcal{D}}$ bit at the encoder. Correspondingly, we call the texture and depth decoders as $\Gamma_{\mathcal{T}} : \{1, 2, \dots, 2^{R_{\mathcal{T}}}\} \to \mathcal{T}$ and $\Gamma_{\mathcal{D}} : \{1, 2, \dots, 2^{R_{\mathcal{D}}}\} \to \mathcal{D}$. Finally, we denote the rendering scheme as $\Upsilon : \mathcal{F} \to \mathcal{G}$. Each rendering scheme has two parts: first, the projection into intermediate view using a few closer reference views and their associated depth maps and second, filling the holes that are not covered by any of these reference views. In this paper we are using only the two closest reference views for rendering. Furthermore, we assume in our theoretical analysis that we have no hole in the reconstructed images. Thus, rendering becomes a simple projection of the closer reference views on an intermediate view using depth information.

Let us denote the decoded data as $\hat{f} = \Gamma_R(\mathcal{E}_R(f))$. The distortion in the rendered version of the data, $\hat{g} = \Upsilon(\hat{f})$, and the original version, $g = \Upsilon(f)$, is given by[2]

$$D(g, \hat{g}) = \sum_{i=0}^{p-1} \|t_i - \hat{t}_i\|_2 + \sum_{j=0}^{q-1} \|v_j - \hat{v}_j\|_2. \tag{2}$$

We finally define the distortion of the coding scheme as the distortion of the encoding algorithm in the least favorable case, i.e.,

$$D_{\mathcal{E},\Gamma,\Upsilon}(R) = \sup_{g \in \mathcal{G}} D(g, \hat{g}). \tag{3}$$

When the encoding, decoding and rendering strategies are clear from the context we use a simpler notation $D(R)$ and call it the distortion-rate (DR) function.

3 Theoretical analysis

In this section we propose a DR function based on our simple model of scenes $\mathcal{H}^Q(\boldsymbol{\Omega})$. We first consider a simple camera configuration $\mathcal{B}_1^1(\mathcal{P})$ with only one reference view and one virtual view. Then we extend analysis to more virtual views with camera configuration $\mathcal{B}_q^1(\mathcal{P})$ and to more reference views with configuration $\mathcal{B}_q^p(\mathcal{P})$. For each class of functions the RD analysis is

[2] In this paper we consider the ℓ_2 distortion. However extensions to other norm losses is straightforward.

built in the wavelet domain where the distortion is the distance between the original and coded wavelet coefficients. The distortion in wavelet domain is equal to the distortion in the signal domain when wavelets form an orthonormal basis, while such a sparse representation of our virtual and reference views simplifies the RD analysis. Assuming that coding has negligible effect on the average signal value, then we can ignore the distortion in the lowest frequency band. Therefore, in the following analysis we only focus on the distortion in high frequency band coefficients. In all the proofs, we assume that the wavelets have a finite support of length ℓ and that their first moments are equal to zero.

Theorem 1 *The coding scheme that uses wavelet-based texture coder and uniform quantization depth coder, achieves the following DR function on scene configuration $\mathcal{H}^1(\boldsymbol{\Omega})$ and camera setup $\mathcal{B}_1^1(\mathcal{P})$*

$$D(R_t, R_d) \sim O(2\mu\sigma^2 2^{\alpha R_t} + K\frac{\varDelta Z}{Z_{min}[2^{\beta R_d} + \varDelta Z]}),$$

where R_t and R_d are the texture and depth bit rates, $K = A'R'|T - T'|$ depends on camera parameters, $\varDelta Z = Z_{max} - Z_{min}$, Z_{max} and Z_{min} are the maximum and minimum depth values in the scene, σ^2 is the reference frame variance and μ, α and β are positive constants.

Proof For the camera configuration $\mathcal{B}_1^1(\mathcal{P})$ we have $g = \{(t_0, v_0)\}$ with one reference view and one virtual view. In all proofs we consider that there is no occluded region for the sake of simplicity. Inspired by [32], we consider the same quantization level for each wavelet coefficient. This suboptimal choice of quantization will only affect constant factors of the DR function and will not change the final upper bound equation. In addition to this, for depth map coding, we assume a quantization-based coder that simply splits depth image into uniform square areas and for each square the average depth is quantized and coded. Therefore, if we assign b bits for coding each wavelet coefficient in the reference frame and b' bits for coding each depth value, there will be three sources of distortion after decoding and rendering at the decoder side,

First at every scale s the number of non-zero wavelet coefficients is $3 \times d\varOmega\ell 2^s$ where $d\varOmega$ is the boundary length of $\varOmega$ in v_0 and 3 factor is because of three wavelet bands. Using the definitions of section 2.1, the magnitude of coefficients at scale s of a standard wavelet decomposition is bounded by

$$|C^1_{s,n_1,n_2}| \leq \int_{t_0}^{t_0+\ell 2^{-s}} \int_{t'_0}^{t'_0+\ell 2^{-s}} |f(t_1,t_2)||\varPsi^1_{s,n_1,n_2}(t_1,t_2)|dt_1dt_2 \leq 2^s \int_{t_0}^{t_0+\ell 2^{-s}} \int_{t'_0}^{t'_0+\ell 2^{-s}} |\phi(2^s t_1 - n)\psi(2^s t_2 - n)|dt_1dt_2 \leq 2^{-s}.$$

We have similar results in case of $|C^2_{s,n_1,n_2}|$ and $|C^3_{s,n_1,n_2}|$. By assigning b bits for coding each coefficient, clearly all the coefficients at scale s, $2^{-s} < 2^{-b-1}$, will be mapped into zero. Therefore, the first source of coding distortion D_1 is

$$D_1 = 3\ell d\varOmega \sum_{s=b+2}^{\infty} 2^s \times (2^{-s})^2 = c_1 2^{-b} \tag{4}$$

where $c_1 = 12\ell d\varOmega$. Note that a factor of 2 is added here because the error of skipping small wavelet coefficients affects distortion in both t_0 and v_0 similarly.

Then, depth map quantization also introduces distortion as it leads to shifts in foreground objects. Recall that we are calculating distortion in the wavelet domain. Consider s_1 as the largest scale with wavelet support length that is smaller than the amount of shift in foreground object. Non-zero wavelet coefficients at scales larger or equal to s_1 suffer from position changes due to depth coding. Assume $\varDelta_0$ as the maximum position error in v_0 with a b' bits quantization-based depth coder. Then we have $\ell 2^{-s_1-1} < \varDelta_0 < \ell 2^{-s_1}$. Hence, our second source of error, D_2, is

$$D_2 = 2 \times 3\ell d\varOmega \sum_{s=s_1+1}^{b+1} 2^s (2^{-s})^2 = c_1(2^{-s_1} - 2^{-b-1}). \tag{5}$$

Here the factor 2 is due to shift of significant coefficients.

Finally distortion is generated by quantization of non-zero coefficients. Using the definition of b and s_1, for the reference frame, t_0, we have large coefficients quantization error in $s \leq b+1$ and for virtual view, v_0, this happens at $s \leq s_1$. Thus, for the last source of distortion we have

$$\begin{aligned} D_3 &= 3\ell d\varOmega [\sum_{s=1}^{b+1} 2^j (2^{-b-1})^2 + \sum_{s=1}^{s_1} 2^s (2^{-b-1})^2] \\ &= c_1(2^{-b} + 2^{s_1} 2^{-2b}). \end{aligned} \tag{6}$$

Using (4), (5) and (6) the total distortion is

$$D = c_1[2^{-b} + 2^{-s_1} - 2^{-b-1} + 2^{-b} + 2^{s_1}2^{-2b}].$$

From s_1 and $\varDelta_0$ definitions we have $s_1 \leq b$ and $s_1 \geq \log \varDelta_0^{-1} - 1$. Therefore, we can simplify the above equation as

$$D = O(2^b + \varDelta_0).$$

The first term only depends on texture coding errors and the second term on depth quantization. We replace the texture coding term with a simple distortion model $\mu\sigma^2 2^{-\alpha R}$ [33] where μ and α are model parameters, σ^2 is the source variance and R is the target bit rate. Using the formulation of maximum shift error in [21] for the depth distortion term we finally have

$$D(R_t, R_d) = O(2\mu\sigma^2 2^{-\alpha R_t} + A'R'|T - T'|\frac{Z_{max} - Z_{min}}{Z_{min}[2^{\beta R_d} + Z_{max} - Z_{min}]}) \quad (7)$$

where β is another model parameter that depends on depth coding method.

We now extend the above analysis to more complex camera configurations. We first consider q virtual views in a $\mathcal{B}_q^1(\mathcal{P})$ configuration.

Theorem 2 *The coding scheme that uses wavelet-based texture coder and uniform quantization depth coder, achieves the following DR function on scene configuration $\mathcal{H}^1(\mathbf{\Omega})$ and camera setup $\mathcal{B}_q^1(\mathcal{P})$*

$$D(R_t, R_d) \sim O((q+1)\mu\sigma^2 2^{\alpha R_t} + \sum_{j=0}^{q-1} K_j \frac{\varDelta Z}{Z_{min}[2^{\beta R_d} + \varDelta Z]}),$$

where R_T and R_D are the texture and depth rates, $K_j = A'_j R'_j |T - T'_j|$, for $j = 0, \ldots, q-1$ depends on camera parameters, $\varDelta Z = Z_{max} - Z_{min}$, Z_{max} and Z_{min} are the maximum and minimum depth values in the scene, σ^2 is the reference frame variance and μ, α and β are positive constants.

Proof With q virtual cameras the three sources of distortion in the proof of Theorem 1 turn into

$$D_1 = c_1(q+1)2^{-b}, \quad (8)$$

$$D_2 = 2 \times 3\ell d\varOmega \sum_{j=0}^{q-1} \sum_{s=s_j+1}^{b+1} 2^s(2^{-s})^2 = c_1(\sum_{j=0}^{q-1} 2^{-s_j} - q2^{-b-1}) \quad (9)$$

and

$$D_3 = c_1(2^{-b} + 2^{-2b}\sum_{j=0}^{q-1} 2^{s_j}). \quad (10)$$

We have $s_j \leq b$ and $s_j \geq \log \varDelta_j^{-1} - 1$ for $j = 0 \ldots q-1$, thus

$$D = O((q+1)2^b + \sum_{j=0}^{q-1} \varDelta_j).$$

Here, we have simply used the fact that the error in the virtual views augments with the number of such views. The DR function is then obtained by following exactly the same replacements as in the proof of Theorem 1.

Finally we extend the analysis to configurations with more reference views. We assume that we have equally spaced reference cameras and virtual views, and that the number of intermediate views is uniform between every two reference cameras. A weighted interpolation strategy using the two closest reference views is employed for synthesis at each virtual view point. The weights are related to the distances between corresponding virtual view and right and left reference views similarly to [19]. Theorem 3 provides the general DR function in a general camera configuration with p reference views and $(p-1)q$ virtual views.

Theorem 3 *The coding scheme that uses wavelet-based texture coder and uniform quantization depth coder, achieves the following DR function on scene configuration $\mathcal{H}^1(\mathbf{\Omega})$ and camera setup $\mathcal{B}_q^p(\mathcal{P})$*

$$D(R_{t_0}, \ldots, R_{t_{p-1}}, R_{d_0}, \ldots, R_{d_{p-1}}) \sim O(\sum_{i=0}^{p-1} \mu\sigma_i^2 2^{\alpha R_{t_i}} + \sum_{j=0}^{(p-1)q} (\frac{d_{j,r}}{d})^2[\mu\sigma_l^2 2^{\alpha R_{t_l}} + K_{j,l}\frac{\varDelta Z}{Z_{min}[2^{\beta R_{d_l}} + \varDelta Z]}] + (\frac{d_{j,l}}{d})^2[\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r}\frac{\varDelta Z}{Z_{min}[2^{\beta R_{d_r}} + \varDelta Z]}]),$$

where R_{t_i} and R_{d_i} are the texture and depth rates for the ith reference view, $\varDelta Z = Z_{max} - Z_{min}$, Z_{max} and Z_{min} are the maximum and minimum depth values in the scene, σ_i^2 is variance of the ith reference view and μ, α and β are positive constants. Also, d indicates the distance between each two reference cameras and $d_{j,l}$ and $d_{j,r}$ are the distances between jth virtual view and *its left and right reference cameras. Similarly, we have* $K_{j,l} = A'_j R'_j |T_l - T'_j|$ *and* $K_{j,r} = A'_j R'_j |T_r - T'_j|$ *that depend of camera parameters.*

Proof First, using Theorem 2, we can write the distortion of a reference view, r, and the q virtual views on its left as

$$D(R_{t_r}, R_{d_r}) = O(\mu\sigma_r^2 2^{\alpha R_{t_r}} + \sum_{j=0}^{q-1} [\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r} \frac{\Delta Z}{Z_{min}[2^{\beta R_{d_r}} + \Delta Z]}]) \tag{11}$$

Clearly, the first and second terms define the distortion at reference and virtual views, respectively. By adding another reference view, l, and using a weighted average of the two closest reference views for synthesizing virtual views we have

$$\begin{aligned} D(R_{t_r}, R_{t_l}, R_{d_r}, R_{d_l}) = \\ O(\mu\sigma_r^2 2^{\alpha R_{t_r}} + \mu\sigma_l^2 2^{\alpha R_{t_l}} + \\ \sum_{j=0}^{q-1} (\frac{d_{j,r}}{d})^2 [\mu\sigma_l^2 2^{\alpha R_{t_l}} + K_{j,l} \frac{\Delta Z}{Z_{min}[2^{\beta R_{d_l}} + \Delta Z]}] + \\ (\frac{d_{j,l}}{d})^2 [\mu\sigma_r^2 2^{\alpha R_{t_r}} + K_{j,r} \frac{\Delta Z}{Z_{min}[2^{\beta R_{d_r}} + \Delta Z]}]) \end{aligned} \tag{12}$$

where d indicates the distance between the two reference cameras and $d_{j,l}$ and $d_{j,r}$ are the distances between jth virtual view and its left and right reference cameras. Here, our weights are simply related to the distance between virtual view and its neighbor reference views. Finally, summing up the terms of (12) for all reference views, leads to the distortion in Theorem 3.

The above rate-distortion analysis is performed on $\mathcal{H}^1(\mathbf{\Omega})$. However, the extension to multiple foreground objects is straightforward and only adds constant factors to the RD function.

4 RD Optimization

In this section we show how the analysis in Section 3 can be used for optimizing the rate allocation in multiview video coding. Using Theorem 3, the rate allocation problem turns into the following convex nonlinear multivariable optimization problem with linear contraints

$$\begin{aligned} & arg \min_{\overrightarrow{R}_t, \overrightarrow{R}_d} g_t(\overrightarrow{R}_t) + g_d(\overrightarrow{R}_d) \\ & \text{such that } \|\overrightarrow{R}_t + \overrightarrow{R}_d\|_1 \leq R \end{aligned} \tag{13}$$

where

$$g_t(\overrightarrow{R}_t) = \sum_{i=0}^{p-1} (q+1)\mu\sigma_i^2 2^{\alpha R_{t_i}},$$

$$\begin{aligned} g_d(\overrightarrow{R}_d) = \sum_{j=0}^{(p-1)q} [(\frac{d_{j,l}}{d})^2 K_{j,r} \frac{\Delta Z}{Z_{min}[2^{\beta R_{d_r}} + \Delta Z]} + \\ (\frac{d_{j,r}}{d})^2 K_{j,l} \frac{\Delta Z}{Z_{min}[2^{\beta R_{d_l}} + \Delta Z]}] \end{aligned}$$

and R is the total target bit rate. The convexity proof is straightforward since the above optimization problem is sum of terms in the form $a2^{-bx}$, which are convex. Therefore it can be solved efficiently using classical convex optimization tools. Note that the above optimization problem is for the general camera configuration $\mathcal{B}_q^p(\mathcal{P})$. The rate allocation for simpler configurations is straightforward by replacing the objective functions with terms from Theorem 1 and 2. We can finally note that the rate allocation strategy is only based on encoder side data.

The last issue that we have to address is adjustment of the model parameter. There are three parameters, μ, α and β in (13) that we estimate using the following offline method. Using the first texture and depth images, we estimate the model parameters by solving the following regression

$$\min_{\mu,\alpha,\beta} \sum_{k=0}^{n-1} |D(R_k) - D^*(R_k)| \tag{14}$$

where n is the number of points in the regression and is further discussed in the next section and $D(R_k)$ is the distortion obtained by our rate allocation strategy of Eq. (13) with target bit rate R_k and $D^*(R_k)$ is the best possible allocation obtained by a full search method at the same bit rate.

5 Experimental Results

In the previous sections, we have studied the bit allocation problem on simple scenes and extracted a model for estimating RD function of a DIBR multiview coder with wavelet-based texture coding and a quantization-based depth coding. This section studies the RD behavior and the accuracy of proposed model on real scenes where JPEG2000 is used for coding depth and reference images.

We use the *Ballet* and *Breakdancers* datasets from Interactive Visual Group of Microsoft Research [34]. In our simulations gray-scale versions of these datasets are used. These datasets contain 100 frames and all the numerical results in this section are the average on the three frames from beginning, middle and end of these sequences, i.e., frames with temporal indices 0, 49 and 99. The camera intrinsic and extrinsic parameters, $\mathcal{P}$, and the scene parameters, Z_{min} and Z_{max}, are set to the values given by datasets. In cases where parameters are changed to study the model under some special aspects, we mention the parameter values explicitly.

In an offline stage using Eq. (14) we adjust μ, α and β parameters in Eq. (13) at four bit rates, i.e., $n = 4$, for each dataset. The parameter values are set to $(0.9, 20.5, 8.5)$ for *Ballet* and $(0.9, 30.0, 2.7)$ for *Breakdancers*. These values are fixed all over this section for the different camera configurations.

In the following sections we study the RD model of Eq. (13) for rate allocation in different camera configurations, $\mathcal{B}_1^1(\mathcal{P})$, $\mathcal{B}_6^1(\mathcal{P})$ and $\mathcal{B}_3^2(\mathcal{P})$. As a comparison criterion we use the optimal allocation that is obtained by rendering all the intermediate views and searching the whole distortion-rate space for the allocation with minimal distortion.

As we want to keep our model independent of any special strategy for filling occluded regions, all occluded regions are ignored in distortion and PSNR calculations.

5.1 $\mathcal{B}_1^1(\mathcal{P})$ configuration

We start with $\mathcal{B}_1^1(\mathcal{P})$ camera setup, a simple configuration with one reference view and only one virtual view. As reference and target cameras, we use the cameras 0 and 1 of the datasets, respectively. Thus, all camera-related parameters in Eq. (13) are set accordingly.

A DR surface is first generated offline for the desire bit rate range to generate the distortion benchmark values. In our study, R_t and R_d are set between 0.02 and 0.5 bpp with 0.02 bpp steps. It means that R_t and R_d axes are discretized into 25 values. Since the images are gray and we are coding only one reference view and one depth map, this range of bit rate is pretty reasonable. The DR surface is generated by actually coding the texture and depth images at each (R_t, R_d) pair and by calculating distortion after decoding and synthesis.

Then, for each target bit rate, R, the optimal rate allocation is calculated by cutting the above surface with a plane $R_t + R_d = R$ and minimizing the distortion. If the minimum point occurs between grid points (because we have a discretized surface) bicubic interpolation is used to estimate the optimal allocation. Here, R is set between 0.1 to 0.5 bpp with 0.01 bpp step. Figure 3 provides distortion curves of compression performance of DIBR coder in terms of PSNR for *Ballet* and *Breakdancers* datasets. The estimated curve is generated by solving the optimization problem provided in Eq. (13) with the proposed RD model. The final PSNR results are averaged over frames 0, 49 and 99 of these datasets. The average differences between the model-based and optimal curves are 0.05 dB and 0.06 dB for *Ballet* and *Breakdancers* sets, respectively. Also, the maximum loss in PSNR in our model-based rate allocation is 0.11 and 0.13 dB, respectively.

Table 1 shows the percentage of the total rate that is used for coding texture for different target bit rates. Clearly our model-based allocation follows closely the best allocation. Figure 4 further shows the best and model-based allocations versus bit rate in terms of R_t percentage. Additionally, two dotted curves are presented which are the higher and lower bounds on R_t allocation where the PSNR loss compared to the best allocation remains below 0.2 dB.

We study now the performance of a priori given rate allocations, which are commonly adopted in practice. We consider several such allocations, where the values of R_t relative to the total budget spans a range of 20 to 80 %. Table 2 shows the average PSNR loss compared to the best allocation in these cases. All these results are the average over frames 0, 49 and 99 in both datasets. We compare the performance of the rate allocation estimated with our RD model and we show that our allocation is always better. Figure 4 further shows that using a model-based allocation instead of a priori allocation is more important at low bit rates or in images with close to camera objects (like *Ballet*). Depending on the dataset, the best a priori allocation occurs at different R_t percentages. In our proposed allocation, the results are close to optimal in both datasets as the model adopts to the scene content. The last two rows of Table 2 shows the average benefit of our model compared to a fixed rate allocation.

Finally we study the effect of the distance of virtual views on the rate allocation. We vary the distance between reference and virtual view from 1 to 20 cm by only changing the value of the x coordinate in the T' translation vector of the virtual camera. We further fix the total bit rate to $R = 0.24$ bpp. Figure 5 shows the best rate allocation as a function of the distance of the virtual view. Again, these results are the average over frame 0, 49 and 99 of *Ballet* and *Breakdancers* datasets. Intuitively, for a given error in depth maps due to coding effects, rendering distortion should be smaller in closer virtual views than farther ones. It means that for rendering far views we need more accurate depth information for rendering far views. Alternatively, texture coding distortion plays a more important role in closer

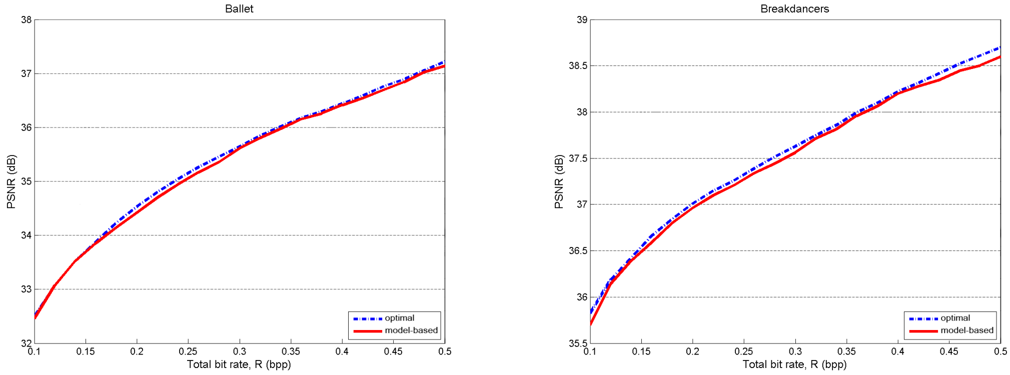

Fig. 3 Comparison of coding performance for $\mathcal{B}_1^1(\mathcal{P})$ using the proposed allocation method and the best allocation in terms of PSNR at rates ranging from 0.1 to 0.5 bpp; *Ballet* (left) and *Breakdancers* (right).

Table 1 Rate Allocation Results for $\mathcal{B}_1^1(\mathcal{P})$ - Comparison between allocation with the proposed model and the optimal allocation, in terms of R_t percentage of the total rate.

Total bitrate		0.2 bpp	0.3 bpp	0.4 bpp	0.5 bpp
Ballet	optimal	67.83%	57.78%	53.33%	37.33%
	model-based	54.61%	49.61%	48.46%	48.36%
Breakdancers	optimal	80.91%	75.56%	70.32%	73.33%
	model-based	75.72%	74.54%	75.09%	75.78%

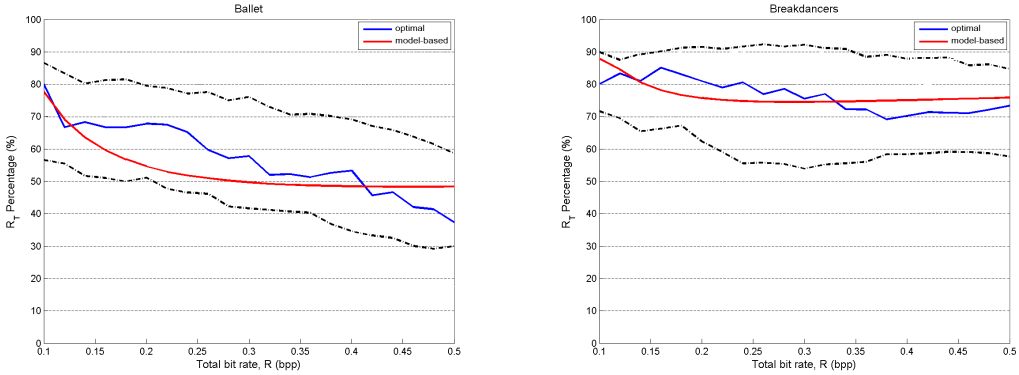

Fig. 4 Rate allocation results of $\mathcal{B}_1^1(\mathcal{P})$ using our proposed method and the optimal allocation in terms of R_t percentage of total rates ranging from 0.1 to 0.5 bpp; *Ballet* (left) and *Breakdancers* (right). The black dashed curves show the bounds within which the difference in PSNR quality with optimal allocation remains less than or equal to 0.2 dB.

views. This is shown in Figure 5 as the R_t percentage decreases by increasing the distance of the virtual view. For *Ballet* dataset we however observe an increase in R_t after 12 cm. It is due to the nature of this scene and to the fact that we use only one camera for rendering virtual views. In this sample there are two foreground objects which are close to the camera, and, beyond a given distance, they move out of view boundaries and mostly background pixels remain. Clearly depth coding errors is less important for background regions that are far from the camera. We also show in Figure 5 the model-based allocation using our RD equation in Eq. (13) where we only change T'. Therefore, the second of the distortion grows with the distance which means that increasing R_d yields smaller distortion comparing to increasing R_t. The average PSNR penalty of our model-based allocation is 0.05 dB and 0.03 dB for *Ballet* and *Breakdancers*, respectively.

5.2 $\mathcal{B}_q^1(\mathcal{P})$ configuration

In this section we study the allocation problem for camera configuration with multiple virtual views. The camera 4 of *Ballet* and *Breakdancers* datasets is used as the reference camera and six virtual cameras separated by 1 cm are considered, three at each side of the reference camera. At each side the parameters of the virtual cameras are set according to camera 3 and 5, respectively.

Table 2 Performance penalty for fixed allocation in $\mathcal{B}_1^1(\mathcal{P})$ - Comparison between the proposed model and a priori allocation policies in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.1 to 0.5. The column headers indicate the a priori allocation of R_t relatively to the total rate.

R_t percentage		20%	30%	40%	50%	60%	70%	80%	our model
Ballet	Average (dB)	1.43	0.66	0.29	0.11	0.07	0.14	0.35	**0.05**
	Maximum (dB)	2.20	1.15	0.63	0.31	0.21	0.39	0.77	0.11
Breakdancers	Average (dB)	1.97	1.14	0.68	0.40	0.21	0.10	**0.06**	**0.06**
	Maximum (dB)	3.23	2.08	1.33	0.77	0.44	0.16	0.11	0.13
Overall	Average (dB)	1.70	0.90	0.49	0.26	0.14	0.12	0.21	**0.06**
	Maximum (dB)	2.72	1.62	0.98	0.54	0.33	0.28	0.49	0.12

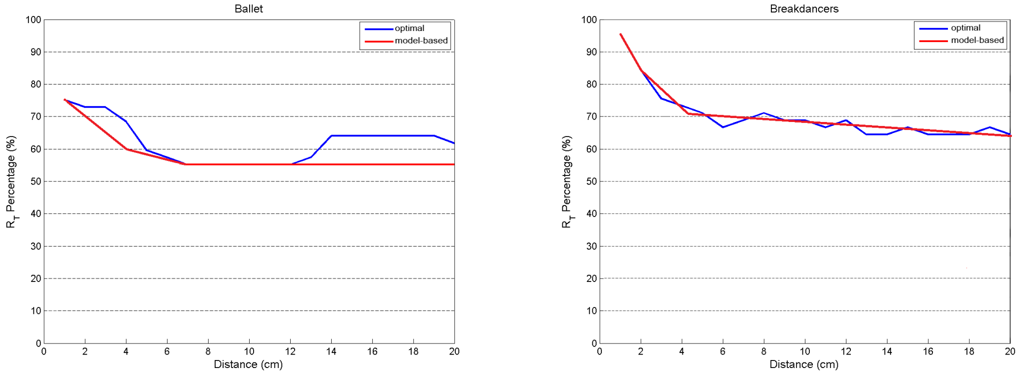

Fig. 5 Rate allocation results of $\mathcal{B}_1^1(\mathcal{P})$ using the model-based and the optimal allocation in terms of R_t percentage at a total rate of 0.24 bpp; *Ballet* (left) and *Breakdancers* (right). The virtual view is projected at 1 to 20 centimeters from reference view.

The optimal allocation process is obtained similarly to section 5.1. The optimal RD surface is generated offline, for R_t and R_d rates between 0.02 and 0.5 bpp with 0.02 bpp steps. Then, at each bit rate R, the best allocation is calculated using interpolation over this RD surface. The model-based allocation is the result of solving Eq. (13) for $\mathcal{B}_6^1(\mathcal{P})$. The reported distortion is the average distortion over all six virtual views and the reference view and also over the three representative frames in each set, i.e., frames 0, 49 and 99.

Figure 6 represents performance in terms of PSNR with respect to target bit rate, R, where R varies between 0.1 and 0.5 bpp. The two curves correspond to the best allocation and the model-based allocation. The amount of loss due using our model is 0.05 and 0.03 dB, on average, for *Ballet* and *Breakdancers*, respectively. Also, the maximum difference is 0.22 and 0.21 dB, respectively. Figure 7 shows the best and model-based allocation in terms of percentage of the total rate allocated to R_t, for different values of R. Clearly our model again performs very close to the optimal allocation. This yields to clear improvements over a priori rate allocation as given in Table 3 in case of $\mathcal{B}_6^1(\mathcal{P})$.

5.3 $\mathcal{B}_q^p(\mathcal{P})$ configuration

We now consider the most general configuration, $\mathcal{B}_q^p(\mathcal{P})$, with two reference cameras ($p = 2$) and three equally spaced virtual views between them ($q = 3$). The cameras 4 and 5 are considered as the two reference views and A'_j and R'_j, $j = 1, 2, 3$, for virtual views are set as the average of intrinsic and rotation matrices of our reference cameras. Each virtual view v_j is generated in two steps. If π is the position of v_j, then each of the reference views are projected into π using depth map information. This step produces $v_{j,r}$ and $v_{j,l}$ as projection results from the right and left cameras, respectively. Next, we have

$$v_j = \frac{d_{j,l}}{d} v_{j,r} + \frac{d_{j,r}}{d} v_{j,l} \qquad (15)$$

where d is the distance between two reference cameras, while $d_{j,l}$ and $d_{j,r}$ are the distances between v_j and the left and right reference cameras, respectively.

The allocation problem in this case consists of distributing the available bit budget between two reference views and two depth maps. For comparison purposes, we calculate a DR hypersurface of the best allocation with R_{t_1}, R_{t_2}, R_{d_1} and R_{d_2} ranging from 0.1 to 0.6 bpp with 0.05 steps. Then for each target bit rate, R, the best allocation is the minimum of the re-

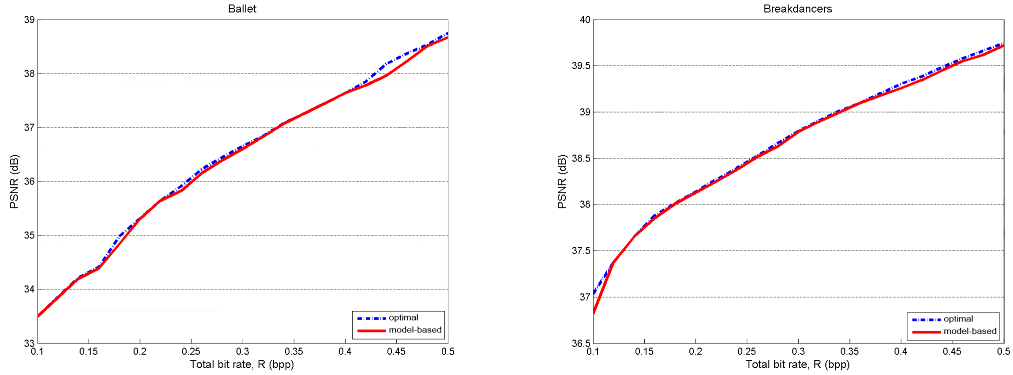

Fig. 6 Comparison of coding performance for $\mathcal{B}_6^1(\mathcal{P})$ using the model-based allocation method and the best allocation in terms of PSNR at rates ranging from 0.1 to 0.5 bpp; *Ballet* (left) and *Breakdancers* (right).

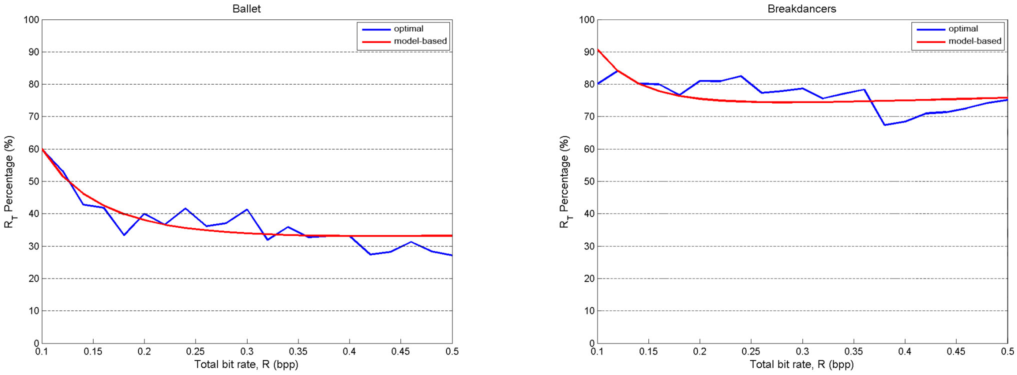

Fig. 7 Rate allocation results of $\mathcal{B}_6^1(\mathcal{P})$ using the model-based and the optimal allocation in terms of R_t percentage at total rates ranging from 0.1 to 0.5 bpp; *Ballet* (left) and *Breakdancers* (right).

Table 3 Performance penalty for fixed allocation in $\mathcal{B}_6^1(\mathcal{P})$ - Comparison between the proposed model and a priori allocation policies in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.1 to 0.5. The column headers indicate the a priori allocation of R_t relatively to the total rate.

R_t percentage		20%	30%	40%	50%	60%	70%	80%	our model
Ballet	Average (dB)	0.54	**0.11**	0.12	0.32	0.67	1.21	1.91	**0.05**
	Maximum (dB)	1.26	0.47	0.35	0.70	1.23	1.90	3.13	0.22
Breakdancers	Average (dB)	2.03	1.15	0.66	0.36	0.16	0.05	**0.03**	**0.03**
	Maximum (dB)	3.57	2.28	1.38	0.78	0.37	0.12	0.08	0.21
Overall	Average (dB)	1.29	0.63	0.39	**0.34**	0.42	0.63	0.97	**0.04**
	Maximum (dB)	3.57	2.28	1.38	0.78	1.23	1.90	3.13	0.22

sulting curve from cutting this hypersurface with the hyperplane $R_{t_1} + R_{t_2} + R_{d_1} + R_{d_2} = R$.

Figure 8 compares the best allocation and the model-based allocation in Eq. (13) for *Ballet* and *Breakdancers* datasets and target bit rates ranging from 0.2 to 0.6 bpp. Our allocation model yields to 0.05 dB loss in average in both cases and a maximum loss of 0.17 and 0.20 dB for *Ballet* and *Breakdancers*, respectively. Figure 9 shows the best and estimated allocations in terms of the percentage of the texture bits $(R_{t_1} + R_{t_2})$ relatively to the total bit rate. The advantage of using our model over the commonly used strategy of a priori rate allocation is shown in Table 4. In the a priori allocation the bit rate assigned to each reference view and depth map is equal. For instance, in $\mathcal{B}_3^2(\mathcal{P})$, if the total bit rate is 0.4 bpp and the a priori allocation is 40%, $R_{t_1} = R_{t_2} = 0.08$ and $R_{d_1} = R_{d_2} = 0.11$ bpp. Clearly our model outperforms the a priori allocation due to adaptivity to content and setup. From Tables 2 to 4, we can conclude that the best performance of an a priori allocation strategy depends on the number of reference and virtual views and on the scene content. While our model-based allocation works well in all cases and gives

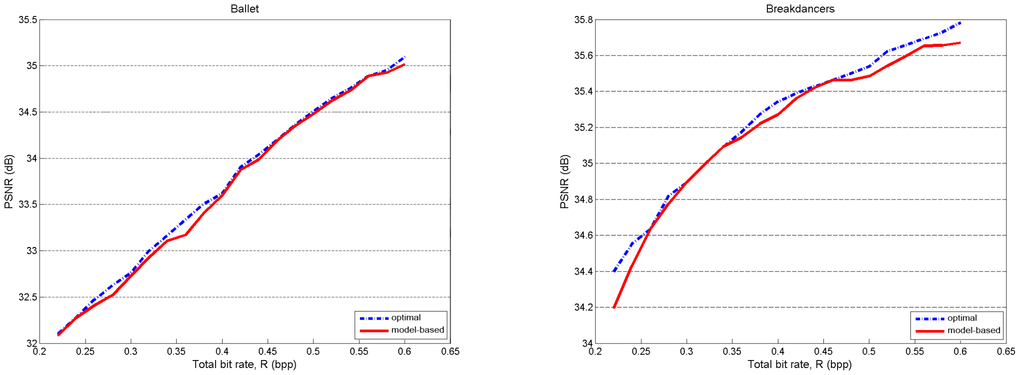

Fig. 8 Comparison of coding performance for $\mathcal{B}_3^2(\mathcal{P})$ using the model-based allocation method and the best allocation in terms of PSNR at rates ranging from 0.22 to 0.6 bpp; *Ballet* (left) and *Breakdancers* (right).

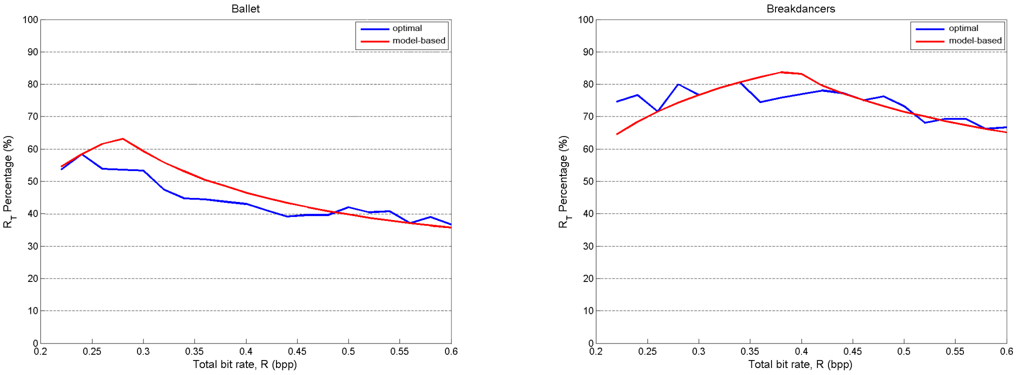

Fig. 9 Rate allocation results of $\mathcal{B}_3^2(\mathcal{P})$ using the model-based and the optimal allocation in terms of R_t percentage at total rates ranging from 0.22 to 0.6 bpp; *Ballet* (left) and *Breakdancers* (right).

Table 4 Performance penalty for fixed allocation in $\mathcal{B}_3^2(\mathcal{P})$ - Comparison between the proposed model and a priori allocation policies in terms of average and maximum differences to the best achievable PSNR at total rates ranging from 0.22 to 0.6. The column headers indicate the a priori allocation of R_t relatively to the total rate.

R_t percentage		20%	30%	40%	50%	60%	70%	80%	our model
Ballet	Average (dB)	0.93	0.26	**0.06**	0.11	0.35	0.68	1.15	**0.05**
	Maximum (dB)	1.49	0.56	0.21	0.24	0.65	1.28	1.79	0.17
Breakdancers	Average (dB)	2.15	1.19	0.68	0.38	0.20	0.11	**0.08**	**0.05**
	Maximum (dB)	3.16	1.95	1.21	0.66	0.47	0.33	0.21	0.20
Overall	Average (dB)	1.54	0.73	0.37	**0.25**	0.28	0.40	0.62	**0.05**
	Maximum (dB)	3.16	1.95	1.21	0.66	0.65	1.28	1.79	0.19

this opportunity to determine number of virtual views later at decoder side.

6 Conclusion

We have addressed the rate-distortion analysis of multiview coding in a depth-image-based rendering context. In particular, we have shown that the distortion in the reconstruction of camera and virtual views at decoder is driven by the coding artifacts in both the reference images and the depth information. We have proposed a simple yet accurate model of the rate-distortion characteristics for simple scenes and different camera configurations. We have used our novel model for deriving effective allocation of bit rate between reference and depth images. One of the interesting features of our algorithm, beyond its simplicity, consists in avoiding the need for view synthesis at encoder, contrarily to what is generally used in state-of-the-art solutions. We finally demonstrate in extensive experiments that our simple model nicely extends to complex multiview scenes with arbitrary numbers of reference and virtual views. It leads to an effective allocation of bit rate with close-to-optimal quality under various rate constraints.

In particular, our rate allocation outperforms common strategies based on static rate allocation, since it is adaptive to the scene content. Finally, we plan to extend our analysis to multiview video encoding where motion compensation poses non-trivial challenges in rate allocation algorithms due to additional coding dependencies.

Acknowledgements

This work has been partially supported by Iran Ministry of Science, Research and Technology and the Swiss National Science Foundation under grant 200021_126894.

References

1. Z. Zhang. Microsoft kinect sensor and its effect. *IEEE Multimedia*, 19:4–10, 2012.
2. P. Merkle, A. Smolic, K. Muller, and T. Wiegand. Efficient prediction structures for multiview video coding. *IEEE Trans. on Circ. and Syst. for Video Technology*, 17(11):1461–1473, 2007.
3. A. Vetro, T. Wiegand, and G. Sullivan. Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standards. *Proc. of the IEEE*, 99(4):626–642, 2011.
4. K. Müller, P. Merkle, and T. Wiegand. 3D Video representation using depth maps. *Proc. of the IEEE*, 99(4):643–656, 2011.
5. D. Tian, P. Lai, P. Lopez, and C. Gomila. View synthesis techniques for 3D video. *Proc. Int. Symp. on Optics, Imaging, and Instrumentation*, 7443, 2009.
6. C. Fehn. Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV. *Proc. SPIE, Stereoscopic Image Process. Render.*, 5291:93–104, 2004.
7. F. Shao, GY Jiang, M. Yu, and Y. Zhang. Object-based depth image-based rendering for a three-dimensional video system by color-correction optimization. *Opt. Eng.*, 50:047006–047006–10, 2011.
8. K.-J. Oh, S. Yea, and Y.-S. Ho. Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-D video. In *Proc. Picture Coding Symp.*, Chicago, IL, May 2009.
9. C.-M. Cheng, S.-J. Lin, S.-H. Lai, and J.-C. Yang. Improved novel view synthesis from depth image with large baseline. In *Proc. Int. Conf. Pattern Recognit.*, Tampa, FL, Dec. 2008.
10. P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P. H. N. de With, and T.Wiegand. The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communication*, 24:73–88, 2009.
11. M. Maitre and M. N. Do. Depth and depth-color coding using shape-adaptive wavelets. *J. of Vis. Commun. and Image Repr.*, 21:513–522, 2010.
12. A. Sanchez, G. Shen, and A. Ortega. Edge-preserving depth-map coding using graph-based wavelets. In *Proc. Asilomar Conference on Signals, Systems and Computers*, pages 578–582, Los Angeles, CA, Nov. 2009.
13. I. Daribo, C. Tillier, and B. Pesquet-Popescu. Adaptive wavelet coding of the depth map for stereoscopic view synthesis. In *Proc. IEEE Int. Workshop on Multimedia Sig. Proc.*, pages 413–417, Paris, France, Oct. 2008.
14. S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer. Efficient depth map compression exploiting segmented color data. In *Proc. Int. Conf. on Multimedia and Expo.*, pages 1–6, July 2011.
15. ITU-T and ISO/IEC JTC 1. Advanced video coding for generic audiovisual services. TU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005 (including FRExt extension), Version 4: Sep. 2005, Version 5 and Version 6: Jun. 2006, Version 7: Apr. 2007, Version 8: Jul. 2007 (including SVC extension), Version 9: Jul. 2009 (including MVC extension).
16. ISO/IEC JTC1/SC29/WG11. Text of ISO/IEC 14496-10:200X/ FDAM 1 multiview video coding. Doc. N9978, Hannover, Germany, Jul. 2008.
17. E. Ekmekcioglu, V. Velisavljevic, and S. T. Worrall. Content adaptive enhancement of multi-view depth maps for free viewpoint video. *IEEE J. of Selected Topics in Sig. Process.*, 5(2):352–361, 2011.
18. J. Y. Lee, H.-C. Wey, , and D.-S. Park. A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images. *IEEE Trans. on Circ. and Syst. for Video Technology*, 21(12):1859–1868, 2011.
19. Y. Liu, S. Ma, Q. Huang, D. Zha, W. Gao, and N. Zhang. Compression-induced rendering distortion analysis for texture/depth rate allocation in 3d video compression. In *Proc. Data Compression Conference*, pages 352–361, Beijing, China, Mar. 2009.
20. H. T. Nguyen and M. N. Do. Error analysis for image-based rendering with depth information. *IEEE on Image Process.*, 18(4):703–716, 2009.
21. W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila. Depth map coding with distortion estimation of rendered view. *Proc. SPIE Visual Inf. Process. Commun*, 7543:75430B–75430B–10, 2010.
22. B. T. Oh, J. Lee, and D.-S. Park. Depth map coding based on synthesized view distortion function. *IEEE J. of Selected Topics in Sig. Process.*, 5(7):1344–1352, 2011.
23. V. Davidoiu, T. Maugey, B. Pesquet-Popescuy, and P. Frossard. Rate distorsion analysis in a disparity compensated scheme. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 857–860, Paris, France, May 2011.
24. M. Maitre and M. N. Do. Joint encoding of the depth image based representation using shape-adaptivewavelets. In *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, pages 1768–1771, Urbana, IL, Oct. 2008.
25. Q. Wang, X. Ji, Q. Dai, and N. Zhang. Free viewpoint video coding with rate-distortion analysis. *IEEE Trans. on Circ. and Syst. for Video Technology*, 22(6):875–889, 2012.
26. G. Cheung, V. Velisavljevic, and A. Ortega. On dependent bit allocation for multiview image coding with depth-image-based rendering. *IEEE Trans. on Image Process.*, 20(11):3179–3194, 2011.
27. A. Gelman, P. L. Dragotti, and V. Velisavljevic. Multiview image coding using depth layers and an optimized bit allocation. *IEEE Trans. on Image Process.*, 21(9):4092–4105, 2012.
28. D. L. Donoho. Wedgelets: Nearly minimax estimation of edges. *Annals of Stat.*, 27, 1999.

29. E. Le Pennec and S. Mallat. Sparse geometrical image approximation with bandelets. *IEEE Trans. Image Process.*, 14(4):423–438, 2005.
30. A. Maleki, B. Rajaei, and H. R. Pourreza. Rate-distortion analysis of directional wavelets. *IEEE Trans. Image Process.*, 21(2):588–600, 2012.
31. S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, CA, 1997.
32. P.Prandoni and M.Vetterli. Approximation and compression of piecewise smooth functions. *Phil. Trans. Royal Society London*, 357(1760):2573–2591, 1999.
33. T. M. Cover and J. A. Thomas. *Elements of Information Theory (Telecommunications and Signal Processing)*. Wiley, New York, 2006.
34. Sequence Microsoft Ballet and Breakdancers. 2004. [Online]. Available: http://research.microsoft.com/en-us/um/people/sbkang/3dvideodownload.